WEST

Generate Collection | Print

L9: Entry 22 of 60 File: USPT Sep 17, 2002

DOCUMENT-IDENTIFIER: US 6453389 B1
TITLE: Optimizing computer performance by using data compression principles to minimize a loss function

Detailed Description Text (14):
As the CPU 12 makes and processes cache line requests, the prefetcher 16 recognizes patterns in the cache misses and generates predictions of soon-to-be-accessed data. Generally, this works as shown in FIG. 3. As each cache line miss occurs, the prefetcher 16 receives the address miss and computes an address difference for generating a data structure as explained in more detail with reference to FIG. 4. If the prefetcher 16 does not make a prediction at step 200, the prefetcher 16 remains idle at step 210. If the prefetcher 16 makes a prediction at step 200, then at step 215 the prefetcher 16 determines if the predicted cache line is already present in the cache 14 or the prefetch cache 18, or a request for the predicted cache line is already present in the queue 20 or on the bus 22. If neither the predicted cache line nor a request for it is already present, the prefetcher 16 issues a cache line request to the queue 20 at step 220. If a request for the predicted cache line already exists, the cache line is returned from the main memory 24 at step 225 and (provided the CPU 12 did not issue the request) is stored in the prefetch cache 18 at step 230.

Detailed Description Text (143):
Consider a system with a cache line size of 2. Assume the sequence of memory references is the progression 1, 2, 3, 4, . . . The corresponding sequence of cache misses will be 1, 3, 5, . . . since the even numbered addresses are brought in to the cache together with the odd numbered addresses because the cache line size is 2. The rough predictor 350 will receive the sequence 1, 3, 5, . . . , n. The rough predictor 350 will predict the next miss to be x+cache line size or x+2. Since the rough predictor 350 guesses correctly, the address difference is 0 and the sequence of prefiltered data is 1, 0, 0, 0 . . . . (Note that the prefetch engine 330 receives more than just the prefiltered data. It also must receive the cache misses or the rough predictions, else it cannot determine the correct address to prefetch.) The initial 1 in the sequence of prefiltered data is chosen arbitrarily and assumes that, in the absence of data, the fixed predictor 350 will predict 0. Thus, the prefetch engine 330 receives as input relative patterns or relations between addresses rather than absolute addresses.